Query-dependent metric learning for adaptive, content-based image browsing and retrieval

Han, Junwei; McKenna, Stephen

Published in:
IET Image Processing

DOI:
10.1049/iet-ipr.2013.0514

Publication date:
2014

Document Version
Peer reviewed version

Query-Dependent Metric Learning for Adaptive, Content-based Browsing and Retrieval of Images

Junwei Han[1] and Stephen J. McKenna[2]

[1]*School of Automation, Northwestern Polytechnical University, China*

[2]*School of Computing, University of Dundee, Dundee DD1 4HN, UK*

Abstract: An adaptive image browsing system based on high-entropy layouts of retrieved image sets, qualitative user feedback provided by manipulation of such layouts, and on-line learning of image-image distance metrics is presented. User feedback involves query update and arrangement of relevant images in relation to the query to generate qualitative constraints. Simulation results are presented that demonstrate the effectiveness of metric adaptation using a maximal-margin learning algorithm in this context.

Keywords*: Content-based image retrieval, image browsing, relevance feedback, metric learning, preference learning.*

1. Introduction

Most early content-based image retrieval (CBIR) systems relied on pre-defined image-to-image similarity measures. These so-called computer-centric systems were relatively easy to implement but inherent drawbacks limited their performance. Image understanding is highly subjective; each user has different personal intentions and preferences when searching for images or browsing an image collection. These can vary from session to session even if identical queries are posed initially. Relevance feedback was proposed to help to address the limitation of using fixed similarity measures [15]. In relevance feedback, on-line learning of query-dependent similarity measures is performed based on user feedback. When adopting this approach it is often assumed that a user is looking for a category of images and starts with a query example from that category. The user

provides feedback on the categorical relevance of retrieved images. Machine learning methods such as support vector machines [18] or manifold learning [7] are used to refine the similarity measure based on the feedback. In [21] and [22], based on the assumption that relevant users may have similar interests, the log data of previous users' relevance judgments were accumulated and integrated with the current user's relevance feedback for the collaborative learning of similarity metrics. Requiring feedback in terms of category-membership decisions burdens users by forcing them to decide upon a useful categorization of images even if unfamiliar with the database [2]. While this can be appropriate for a category-based search, it is not most appropriate for browsing and retrieval more generally. An example of an alternative approach is the target-based image retrieval system proposed by Cox et al. [2] in which a probability distribution over possible targets is iteratively updated based on user behavior. The probability of an image being the target is increased if it is closer to relevant examples in feature space.

The most common relevance feedback in CBIR is in a binary form where users identify a set of examples as *relevant* and thus identify the others as *irrelevant* [7, 18, 21-22]. However, it is often hard and unnatural for users to make these binary decisions because they may have a large variety of ways to understand images. For instance, we suppose that a user intends to search for images of fashion models. The user can easily identify Figure 1(a) as a relevant example and Figure 1(c) as an irrelevant example. It is not clear, however, whether or not Figure 1(b) is of a fashion model. For such a case, qualitative and relative comparison of images is meaningful and preferred. For example, we could all agree on the relative interpretation that Figure 1(a) is more likely to be of a fashion model than Figure 1(b) and Figure 1(b) is more likely to be of a fashion model than Figure 1(c). Indeed, relevance feedback based on relative information can provide consistent responses from users; it can thus improve the precision of communication between users and CBIR systems.This paper proposes to build a model of relative relevance feedback.

Figure 1 Binary relevance feedback is a restrictive way to express a user's judgments. It is clear that (a) is a fashion model and that (c) is not a fashion model, whereas it is difficult to decide whether or not (b) is a fashion model.

The study of how to learn from relative comparisons is attracting increasing attention (see e.g. [19]). Joachims [11] proposed a ranking-SVM method which converted the learning task to a standard SVM classification task. It was applied to learning from `clickthrough' data for Web search engines. Schultz and Joachims [16] extended this approach to learn distance metrics. Freund et al. [4] presented the RankBoost algorithm. Rank-based distance learning has been used to solve vision problems. Frome et al. [5] proposed a method to learn local image-to-image distance functions for classification by combining relative comparison information and image shape features. Hu et al. [8] explored a multiple-instance ranking approach based on ranking-SVM to order images within each category for retrieval. Lee et al. [12] employed a rank-based distance metric to retrieve images of tattoos. Faria et al. [3] also recently used rank learning for CBIR. Huang et al. [10] applied metric learning from rank correlations to a medical image retrieval application. In [23], Wang et al. proposed a regularized kernel machine algorithm to use comparative object similarity. This learning model can effectively recognize objects with few or no training samples. Recently, Parikh et al. [24] presented a novel methodology for modeling relative visual attributes. It learned a ranking function for each attribute. The learned ranking function can predict the strength of an attribute in an image with respect to other images. Experiments demonstrated that relative attribute prediction had

advantages over binary attribute prediction for two tasks. These methods [5, 8, 12, 3, 10, 23, 24] were investigated under the scenario of *offline* learning.

The ability of a user to provide meaningful feedback depends not only on the quality of the retrieval but also on the manner in which the interface displays retrieved images and facilitates the provision of user feedback. A well-designed interface will make this interaction easy and enjoyable for users, and enhance the efficiency of the system. Rodden [14] performed user studies which demonstrated that 2D layouts could enable users to find a target image or group of images more quickly than when images were arranged in order of similarity to a query image. Moghaddam et al. [13] also argued that visualizing images in a 2D space can be superior, allowing mutual similarities to be reflected. Wang et al. [20] proposed the HELD method for arranging image collections and this is adopted here for display of images in a way that facilitates browsing and user feedback.

This paper describes an interactive image browsing approach that incorporates on-line, rank-based learning using relative information into an efficient interface that visualizes image collections in 2D space and facilitates users to provide relational feedback. In this scenario, a user intends to find his/her envisioned target images. For example, a fashion designer may seek a particular kind of fabric, or a customer may look for a particular style of shoe online. The proposed system basically provides users with target-based image retrieval that is different from traditional category-based retrieval. To achieve this end, we develop an architecture that seamlessly combines image browsing, image retrieval, and relevance feedback. A user starts his/her search session by browsing a collection of images displayed in a 2D layout. He/she may choose a query image perceived to be similar to the envisioned target image. A set of images similar to the query image are returned. The user can offer relational feedback and update the query image to refine results. This relies on a ranking based learning of image relationships.

The proposed work builds on an earlier conference paper (Han et al. [6]) and makes the following contributions. A user feedback mechanism is proposed based on HELD displays in which the user

updates the query and can select and arrange relevant images in relation to it, generating relative feedback. An experimental evaluation is presented based on user simulations that quantifies performance with respect to the method's parameters and in comparison to alternative strategies. It should be noted that algorithms for CBIR are often characterised experimentally by simulating usage based on pre-defined, fixed category labels and deeming retrieved results to be relevant when they share a category label with the query. In contrast, the purpose of the system in this paper is to browse in a non-categorical manner. Therefore, a different evaluation method is proposed.

The rest of this paper is organized as follows. Section 2 describes the proposed browsing and retrieval framework. Section 3 presents evaluation results. Finally, conclusions are drawn in Section 4.

2. Browsing and retrieval framework

2.1 Overview

The iterative browsing framework is summarised in Algorithm 1. Given an image collection, I, an intial query image, q_0, and an initial distance metric, D_0, (typically Euclidean), the N closest matches to the query are retrieved (see Subsection 2.2) and displayed using an automatically generated 2D layout, L_0 (see Subsection 2.3}). Note that if an initial query was not available, an initial layout could be generated based on a representative (or randomly selected) subset of I of cardinality *N*. A browsing session then consists of a sequence of iterations that continue until the user decides to end the session. For example, if the user had a target image in mind, the session ends once this image is found. At the t^{th} iteration, the user is presented with a 2D image layout L_{t-1}. The user selects a query image, q_t, from this layout. This query image may or may not differ from the previous query, q_{t-1}. The user then selects further relevant images from the layout and rearranges these on the display to qualitatively express judgments about their relative similarities to

the query. The selected images and their implied ordering are used to generate a set of inequality constraints, P_t, automatically (see Subsection 2.4). A learning algorithm then uses the selected query and the constraints to obtain a new distance metric, D_t (see Subsection 2.5). This metric is then used to retrieve the closest matches from the image set and a visualization algorithm produces a new 2D layout from these matches for the user.

Algorithm 1 Interactive browsing

$t = 0$

$\{I, q_0, D_0\} \xrightarrow{\text{matcher and visualizer}} \{L_0\}$

repeat

$t = t + 1$

$\{L_{t-1}\} \xrightarrow{\text{user}} \{q_t, P_t\}$

$\{I, q_t, P_t\} \xrightarrow{\text{learner}} \{D_t\}$

$\{I, q_t, D_t\} \xrightarrow{\text{matcher and visualizer}} \{L_t\}$

Until user ends session

2.2 Retrieving images

The problem of retrieving N images that are closest to a query image q based on high-dimensional feature vectors has been studied extensively. We only note here that efficient algorithms exist for performing these nearest neighbour searches approximately (see e.g. [1]). Let $\mathbf{q}$ and $\mathbf{x}$ denote feature vectors representing a query image and a database image, respectively. The dissimilarity of these two images can be measured using a parameterised Mahalanobis distance metric $D(\mathbf{q}, \mathbf{x}; \mathbf{W}) = \sqrt{(\mathbf{q}-\mathbf{x})^T \mathbf{W} (\mathbf{q}-\mathbf{x})}$. The symmetric matrix $\mathbf{W}$ is positive semi-definite to ensure that D is a valid metric, i.e., $\mathbf{W} \geq 0$. If $\mathbf{W}$ is a diagonal matrix, the distance metric becomes a weighted Euclidean distance:

$$D(\mathbf{q}, \mathbf{x}; \mathbf{w}) = \sqrt{\langle \mathbf{w} \cdot ((\mathbf{q}-\mathbf{x}) * (\mathbf{q}-\mathbf{x})) \rangle} \quad (1)$$

where "$\langle\cdot\rangle$" denotes the inner product and "$*$" the element-wise product of two vectors. $\mathbf{w}$ is the vector consisting of diagonal elements of $\mathbf{W}$. In this case, $\mathbf{w} \geq 0$. It is this parametric family of diagonal metrics which is used in the remainder of this paper.

2.3 Generating 2D layouts

At each iteration of browsing, the retrieved image set is arranged automatically for display using a 2D layout generation algorithm. Figure 2 shows an example of such a layout; the query image is also shown in the upper left corner. In this example, fifty images were retrieved based on color correlogram features [9]. The layout algorithm arranges the images such that the amount of image overlapping is low and images with similar content appear close together. The layouts were generated using a version of the HELD method [20]. HELD generates layouts that conform to the shape of the available display region, approximate the high-dimensional image feature distribution, and make good use of the available layout space. HELD achieves this trade-off by optimisation of an objective function that combines dimensionality reduction with a layout entropy measure. The image set is used to define a density function in the 2D layout space and distributions with low differential Renyi entropy are penalized since they result in layouts in which some regions are over-populated (i.e., many images are occluded) and other regions are sparsely populated or empty. In [20] the ISOMAP algorithm was used within HELD. Here instead multi-dimensional scaling (MDS) was used because of the relatively small number of images in each layout. The distance in feature space between two images was measured using the current browsing metric, D_t.

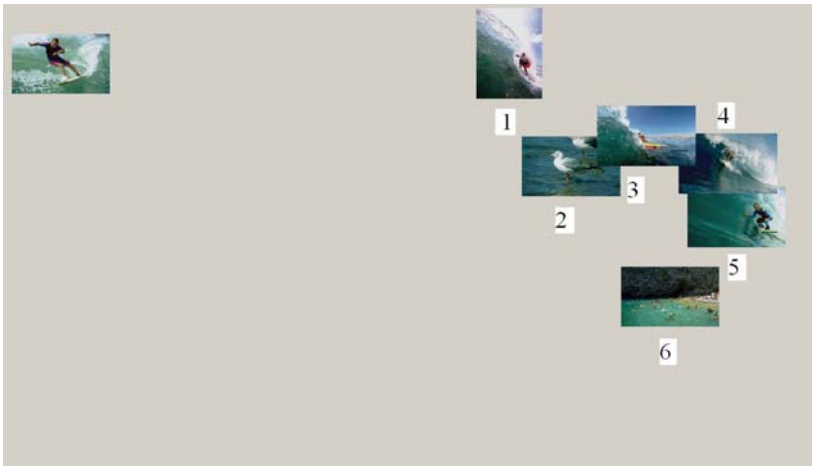
Figure 2: A 50-image HELD layout generated using MDS and Renyi entropy. The query is shown in the upper left corner.

2.4 Obtaining user feedback

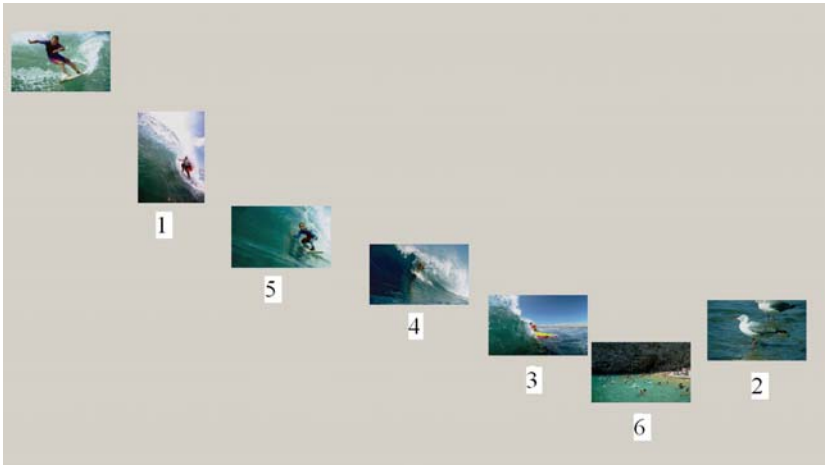
The user might find what they seek in the current 2D layout and decide to terminate the session. If, on the other hand, they decide to continue the session they have the option to select one of the newly retrieved images as a new query. A user might decide to choose a new query either because it matches more closely what they seek or because what they seek has changed, perhaps as a result of inspiration gained from the new layout. The user can then select a subset of the images on the layout as being of relevance to them. Once this selection has been made the other images are removed from the display. Figure 3(a) shows an example in which six retrieved images have been selected from a layout by the user as relevant.

The user can then provide qualitative feedback to the system by arranging the selected images to reflect their perceived relative similarity to the query. Images that are similar to the query are placed close to it and images that are less similar are placed further away. Figure 3(b) shows the example in Figure 3(a) after the user has arranged the selected images. This user-defined layout yields an

ordering of the images in terms of similarity to the query. This ordering implies a set of inequalities on the distance measure being used by the user.



(a)



(b)

Figure 3: (a) Six images have been selected as relevant in addition to a query image. (b) The user has arranged these images to reflect their perceived similarity to the query.

If the user arranges M images relative to the query then there are generally $\frac{M(M-1)}{2}$ inequalities expressing order relationships between these M images. (If the user arranges some images to be equidistant from the query this number will reduce accordingly). However, if we

assume that the user's measure is a metric then most of these order relationships are redundant and only $M-1$ inequalities are needed. In the example shown in Figure 3(b) the constraints would be

$$P_t = \{D_t(q_t,1;\mathbf{w}) < D_t(q_t,5;\mathbf{w}), D_t(q_t,5;\mathbf{w}) < D_t(q_t,4;\mathbf{w}), D_t(q_t,4;\mathbf{w}) < D_t(q_t,3;\mathbf{w}),$$
$$D_t(q_t,3;\mathbf{w}) < D_t(q_t,6;\mathbf{w}), D_t(q_t,6;\mathbf{w}) < D_t(q_t,2;\mathbf{w})\}.$$

Additionally, non-selection of an image by the user can be taken to imply that it is more dissimilar to the query than any selected image. In this case, there are an additional $N-M$ inequalities giving a total of $N-1$ constraints.

2.5 Adapting the metric

The query and the user-generated constraints provide feedback information about the perceptual measures currently being employed by the user. A learning algorithm is used to determine a new distance metric that makes use of this information. Specifically, the objective of the learner is to infer the parameter $\mathbf{w}_t$ of the distance metric $D_t(.,.;\mathbf{w}_t)$. Ideally, this metric should satisfy the constraints P_t. (Henceforth, the subscript t is omitted). This learning task can be performed using a maximal-margin formulation with slack variables amounting to solving the following optimization problem which has the same form as in [5] and [16].

$$\begin{aligned}
&\min_{\mathbf{w},\xi_{(q,i,j)}} \frac{1}{2}\|\mathbf{w}\|^2 + C\sum_{(q,i,j)} \xi_{(q,i,j)} \\
&s.t. \\
&\forall (D(q,i;\mathbf{w}) > D(q,j;\mathbf{w})) \in P: \\
&\quad D^2(q,i;\mathbf{w}) - D^2(q,j;\mathbf{w}) \geq 1 - \xi_{(q,i,j)} \\
&\forall (q,i,j): \xi_{(q,i,j)} \geq 0 \\
&\mathbf{w} \geq 0
\end{aligned} \tag{2}$$

The term $\|\mathbf{w}\|^2$ measures structural loss, $\xi_{(q,i,j)}$ are slack variables, and C is a trade-off parameter. Substituting Eqn. (1) into the first set of constraints in Eqn. (2) yields

$$\langle \mathbf{w} \cdot (\mathbf{d}_{q,i} - \mathbf{d}_{q,j}) \rangle \geq 1 - \xi_{(q,i,j)} \tag{3}$$

where $\mathbf{d}_{q,i} = (\mathbf{q} - \mathbf{x}_i) * (\mathbf{q} - \mathbf{x}_i)$, and $\mathbf{x}_i$ is the feature vector for the i^{th} image. The final constraint $\mathbf{w} \geq 0$ ensures that the learned distance is a metric. Without this constraint, the setting of the optimization would be the same as that of ranking-SVM and standard quadratic programming solvers such as SVM-Light could be used [11]. Ranking-SVM learns a ranking function which is expected to correctly sort the data. In ranking-SVM, elements of the parameter $\mathbf{w}$ can have negative values thus the ranking values can be negative. Although image retrieval can be formulated as a ranking problem using such an approach [8], this approach is unsuitable for query-by-example. This is because ranking-SVM can have negative outputs, implying that some images are more similar to the query than the query itself (since the output for the query will be zero). This point is demonstrated empirically in Section 3.

Frome [5] proposed a custom dual solver that can guarantee non-negativity of $\mathbf{w}$. It is fast enough to be suitable for online learning during real-time browsing. This solver iteratively updates dual variables until convergence:

$$\mathbf{w}^{(t)} = \max\{\sum_{(q,i,j)} \alpha_{(q,i,j)}^{(t)} (\mathbf{d}_{q,i} - \mathbf{d}_{q,j}), 0\} \tag{4}$$

$$\alpha_{(q,i,j)}^{(t+1)} = \min\{\max\{\frac{1 - \langle \mathbf{w}^{(t)} \cdot (\mathbf{d}_{q,i} - \mathbf{d}_{q,j}) \rangle}{\left\| \mathbf{d}_{q,i} - \mathbf{d}_{q,j} \right\|^2} + \alpha_{(q,i,j)}^{(t)}, 0\}, C\} \tag{5}$$

where $0 \leq \alpha_{(q,i,j)} \leq C$ are the dual variables and are initialized to zero. The reader is referred to [5] for implementation details.

3. Evaluation

3.1 Data set

A set of 10,009 images from the Corel dataset was used containing at least 100 images from each of 79 categories. These categories had semantic labels such as tiger, model, and castle. These labels were used only to ensure a varied data set. It should be stressed that these categories have no role to

play in the browsing framework and that no use was made of the category labels by any of the algorithms.

3.2 Simulating the user

Two types of low-level feature were extracted from the images: 36-dimensional color histograms and 18-dimensional texture features based on a wavelet transformation [17]. These were concatenated to give a 54-dimensional feature vector to represent each image.

Quantitative evaluations were performed by simulating use of the browsing system. A fixed distance metric, $D_{user}(\mathbf{q},\mathbf{x};\mathbf{w}_{user})$ based on the image features was used by a simulated user. Each simulated browsing session was initiated by randomly selecting two images from the database, one as query and one as target. In the first iteration, the system retrieved images based on a pre-specified metric $D_0(\mathbf{q},\mathbf{x};\mathbf{w}_0)$ that differed from D_{user}. At each iteration the system retrieved N images. The simulated user then used D_{user} to select the closest retrieved image to the target as the new query. Additionally, the simulated user selected the $M-1$ next closest (most relevant) images and arranged these images in terms of distance to the query. In this way, inequality constraints were generated and used by the learning algorithm to update the distance metric to better approximate D_{user}. Browsing terminated when the target was retrieved or when a maximum number of iterations was reached. Once an image had been retrieved it was excluded from being retrieved in subsequent iterations.

The number of images retrieved (N) and the number of images selected for ranking feedback (M) at each iteration were free parameters. Larger values of N result in a greater choice of query for the subsequent iteration. Larger values of M result in more feedback information per iteration.

3.2.1 Retrieval performance when $N = M$

A first set of experiments investigated performance under the assumption that all images in each layout were deemed relevant and thus arranged by the user, i.e., $N = M$. In each case this generated $N - 1$ constraints at each iteration.

The distance metric was initialised to only use the texture features. In other words, the weights in Eqn. (1) were set to equal, non-zero values for each of the 18 texture features, and to zero for each of the 36 colour features. In contrast, the simulated user used a distance metric in which colour features had equal, non-zero values and texture features had weights of zero.

Performance was measured as the fraction of trials in which the target was retrieved within a given number of iterations. Figure 4 plots the performance measure for different values of $N \in \{5, 10, 20, 30, 40, 50\}$. In each case, 100 sessions were simulated. Browsing was terminated after 50 iterations if still unsuccessful at that point. Success was nearly always achieved in fewer than ten iterations once N reached 20.

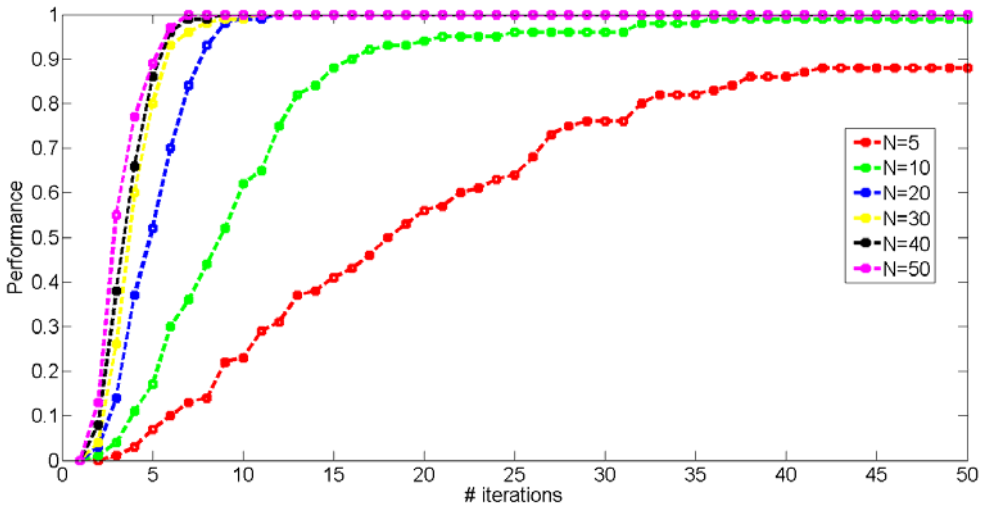


Figure 4: Performance comparisons with different N.

Figure 5 shows the performance obtained without using learning but instead changing the system metric at each iteration by setting the elements of the weight vector $\mathbf{w}$ to 0 or 1 at random. The

retrieved image that was closest to the target using the new metric was selected as the query for the next iteration. The retrieval rates were inferior to those obtained using learning.

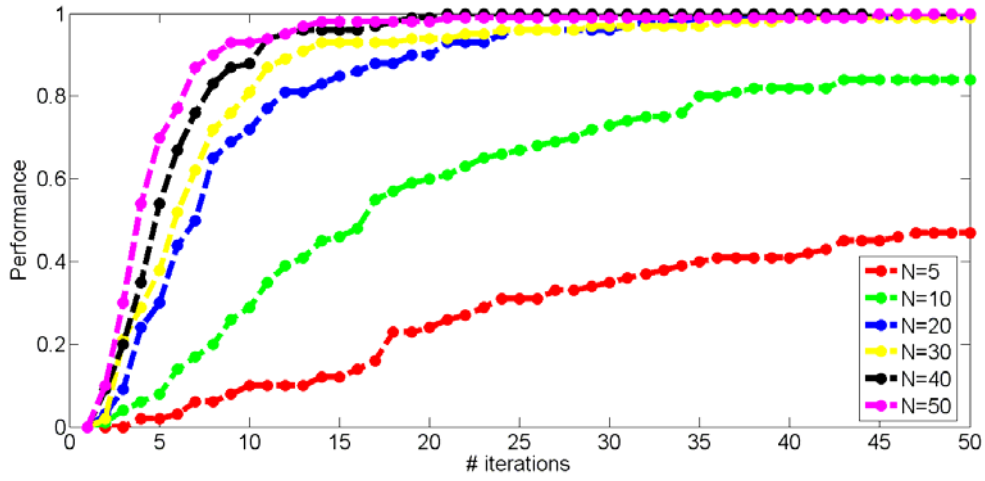
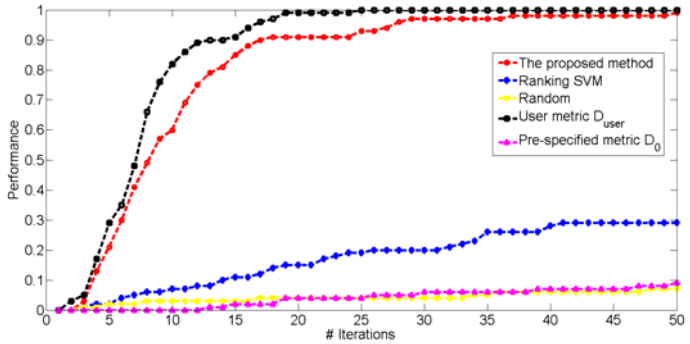
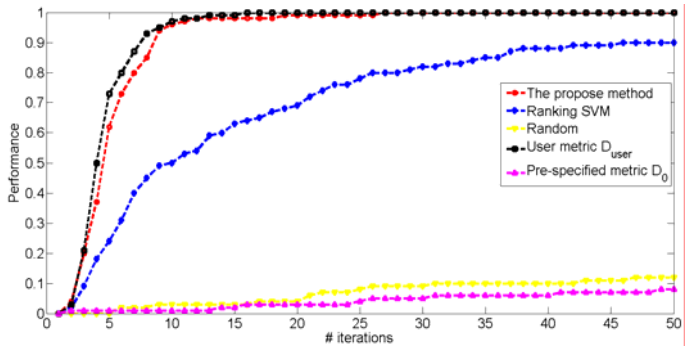


Figure 5: Retrieval performance without learning. The query was reselected and the metric parameter $\mathbf{W}$ was set at random for each iteration.

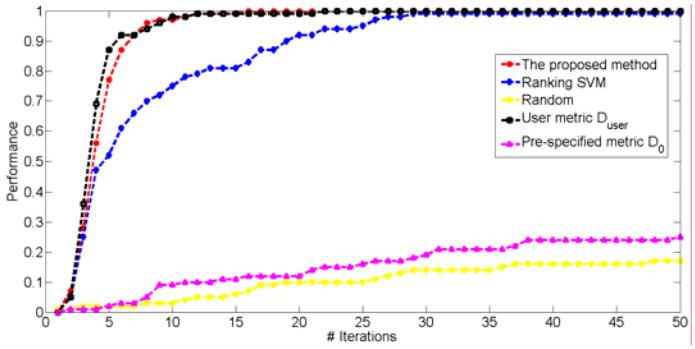
Metric learning was compared to ranking-SVM implemented using code from SVM Light [11]. Two naive methods were also compared. These were random selection of N images without any simulated user interaction, and use of the initial metric for matching throughout. Finally, the methods were compared to retrieval using the ideal metric, D_{user}. Figure 6 shows comparative results for various values of N. The results suggest that the metric learner was superior to ranking-SVM, especially when N was small. For example, for $N=10$, it achieved a retrieval rate of 59% by the tenth iteration, whereas ranking-SVM achieved a rate of 7%. For $N=20$, these rates increased to 96% and 50% respectively. Retrieval rates obtained by metric learning quickly approached those obtained using the ideal metric as N increased. When $N=40$, the rate differed noticeably only for those sessions that succeeded in less than five iterations, and then only by about 5%.



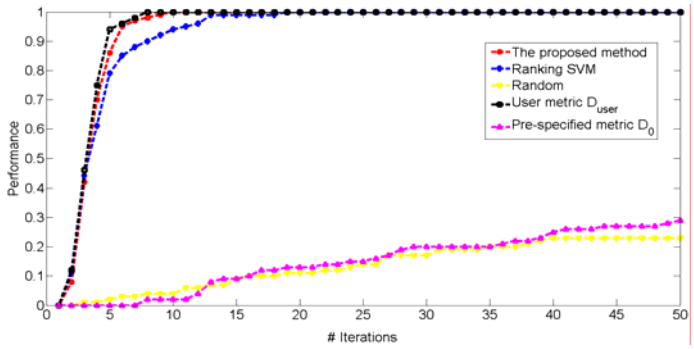
(a) *N=10*



(b) *N=20*

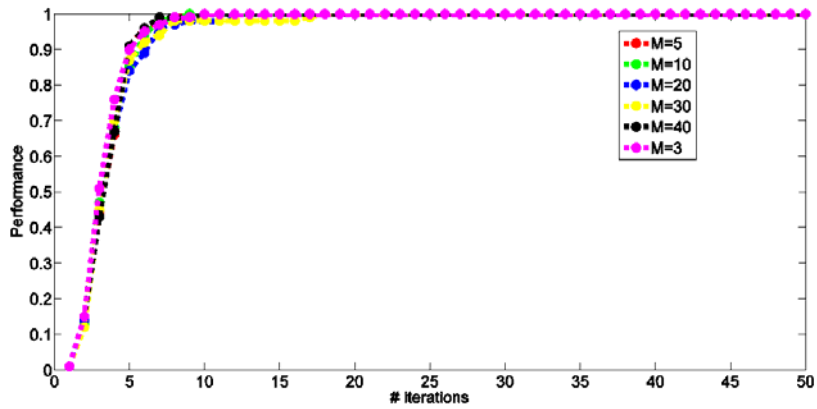


(c) *N=30*

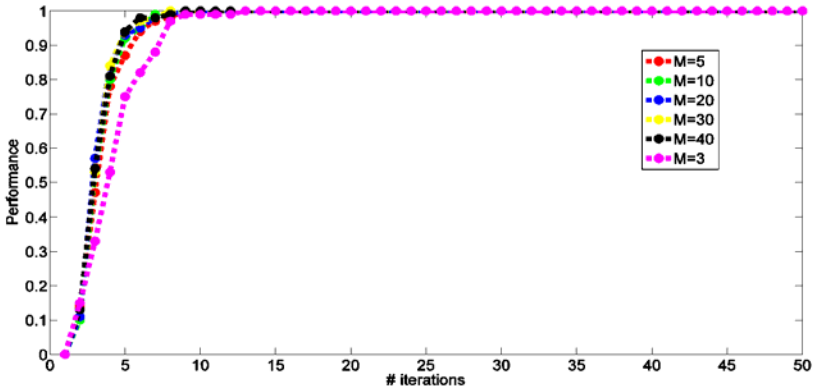


(d) *N=40*

Figure 6: Comparison of retrieval performance of different methods for different values of N .



(a)



(b)

Figure 7: Retrieval performance with $N = 50$ and $M < N$ when using (a) $N - 1$ contrants, and (b) $M - 1$ constraints.

3.2.2 Retrieval performance when $N > M$

A second set of experiments investigated performance when at each iteration not all the images displayed in a layout were selected and arranged by the user, i.e., $M < N$. The simulated user selected those M images that were most similar to the target using the current distance metric, D_t, and arranged them to yield constraints. Figure 7 plots performance for different values of $M \in \{3, 5, 10, 20, 30, 40\}$ when $N = 50$. Figure 7(a) was obtained using all $N - 1$ constraints in each iteration (see Subsection 2.4). Figure 7(b) was obtained using only the $M - 1$ constraints expressing order relationships between the selected images. Comparing these two graphs, the results

suggest that including relationships with non-selected images in the constraint set (i.e., using $N-1$ constraints) is helpful when the number of selected images is small (e.g., $M=3$). These results also demonstrate that the number of iterations is increased only slightly, if at all, when the user selects as few as five images for feedback compared to selecting all images.

3.2.3 Sensitivity to C

An experiment was performed to investigate the effect on retrieval performance of changing the value of the parameter C (see Equation 2). Figure 8 shows results obtained for $C \in \{0.1,1,10,100,1000,10000\}$ when $N=M=20$. As can be seen, changing the value of C over several orders of magnitude had relatively little effect on retrieval rates.

3.2.4 Changing the query

A user might decide to choose a new query image during a browsing session because what they seek has changed, perhaps as a result of inspiration gained from the browsing session so far. However, even when what they seek has not changed, it can be useful to change the query when doing so gives a query that is closer to the `target'. In order to demonstrate the benefit of doing so, the method was run without the ability to change query, i.e., the query image remained fixed throughout a session. Figure 9 shows the result. It can be seen by comparison with Figure 4 that allowing the user to change the query can dramatically reduce the number of iterations needed to successfully complete a session.

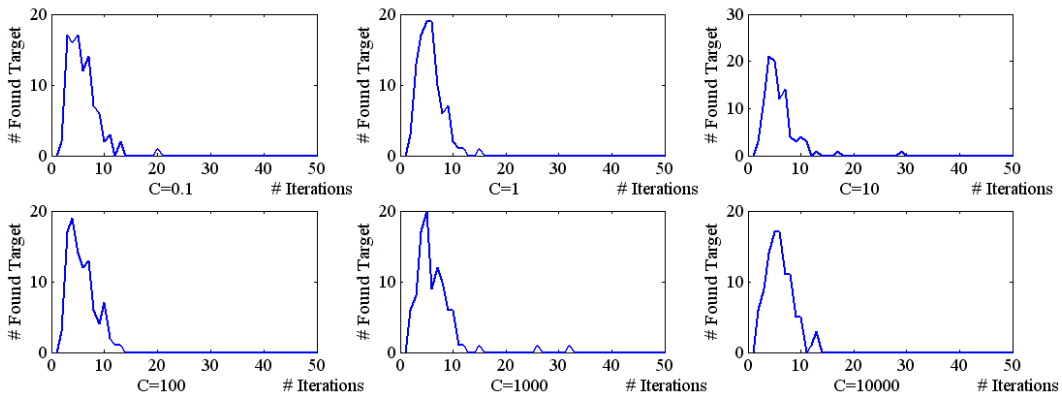


Figure 8: The effect of the value of C. Figures from left to right and from top to bottom are with C set to 0.1, 1, 10, 100, 1000, and 10000, respectively.

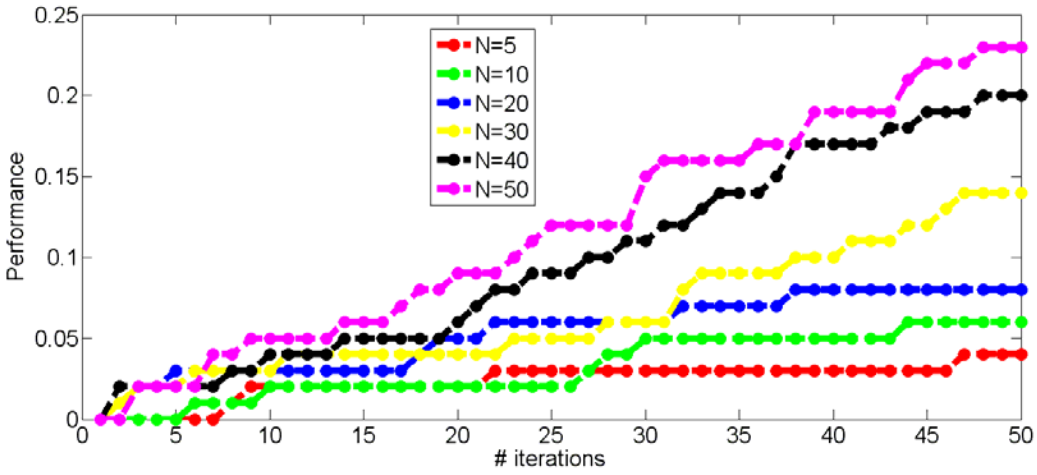


Figure 9: Retrieval performance with query image fixed during each session ($N = M$).

3.2.5 Initial metric mismatch

An experiment was performed to investigate the effect of mismatch between the user metric, D_{user}, and the initial system metric, D_0, on retrieval performance. In this experiment, each element in the parameter vectors $\mathbf{w}_{user}$ and $\mathbf{w}_0$ was set to either 0 or 1, in effect switching features off or on. The Hamming distance between $\mathbf{w}_{user}$ and $\mathbf{w}_0$ was used a measure of the mismatch between the two metrics. Since the feature space was 54-dimensional, the Hamming distance was always in the range [0 54].

Figure 10 plots the mean and standard deviation of the number of iterations per simulated browsing session when $N = M = 20$. For each value of the Hamming distance, 1000 trials were run, each with randomly generated distance metrics having that Hamming distance. No effect of mismatch between user metric and initial system metric is apparent.

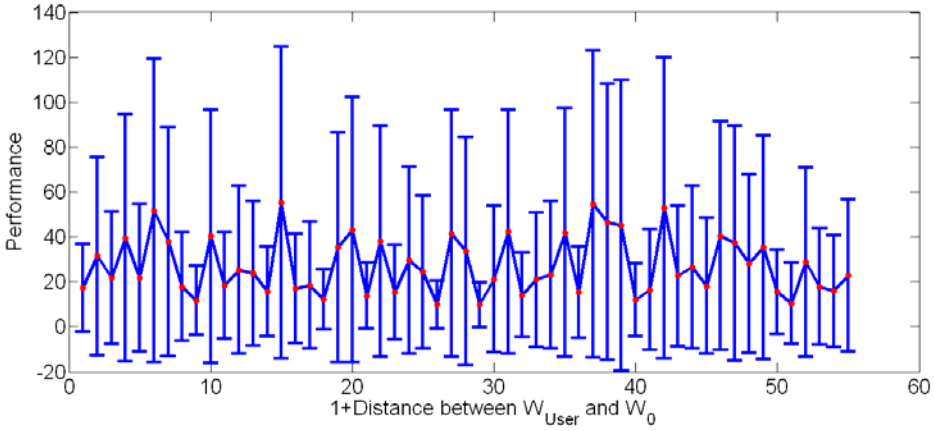


Figure 10: Retrieval performance with different Hamming distances between $\mathbf{w}_{user}$ and $\mathbf{w}_0$.

3.2.6 Distance between query and target

An experiment was performed to investigate the effect of similarity of the query image to the target image. Simulated sessions were run using randomly generated metrics (see Subsection 3.2.5) and randomly selected query and target images. Figure 11 shows a scatter plot from 1000 trials in which the number of iterations is plotted against the distance between initial query and target as measured using D_{user}. The correlation coefficient was -0.22. This suggests that the similarity between initial query and target is not a critical factor in determining retrieval performance.

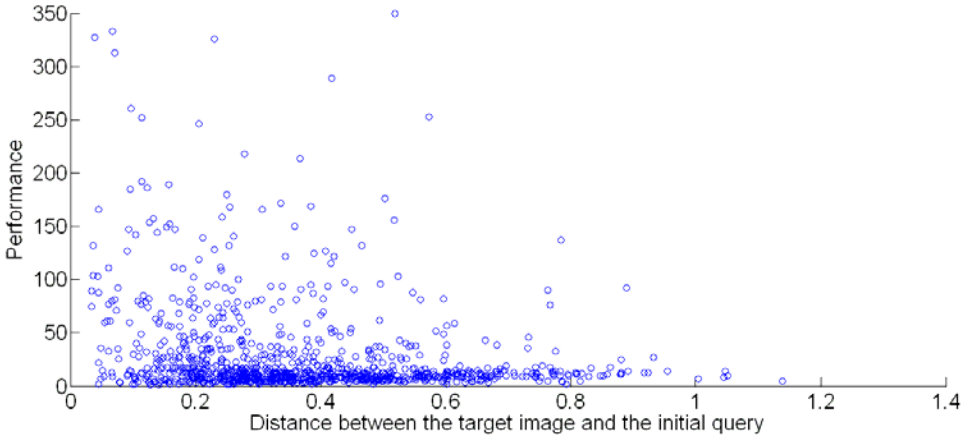


Figure 11: Scatter-plot of performance versus distance between the initial query and target image.

3.3 Interactive online experiment with users

Four subjects (2 male and 2 female) tested the system. Each subject performed ten search sessions. Target images were selected by users and came from 36 different categories. Before each session, the system displayed a layout of 100 images selected randomly from the image database. The user selected whichever of these images was most similar to the target as the initial query image unless the user did not consider any of these 100 images to be similar to the target. In the latter case, the system offered them another 100 randomly selected images from which they were forced to choose. Given the results of the simulation above, $N = M = 20$ was chosen as a reasonable trade-off. A 144-dimensional color correlogram feature vector [9] was used to represent each image in this experiment.

Each iteration requires the user to select a query and move images to provide feedback on similarity to the query. This is more time-consuming than the CPU time for learning, matching and visualization. Query selection normally took less than 10s while arranging the images took 25-50s. If a target was not found after 10 iterations, search was deemed to have failed. There were 40 search sessions in total and, of these, 5 failed, 3 found the target without any interaction other than initial query selection, 20 were successful within 5 iterations, and 12 others were successful using more

than 5 iterations. Overall, successful sessions required an average of 5 iterations to retrieve the target.

4. Conclusions and Recommendations

A framework for an adaptive image browsing system was presented based on 2D HELD visualizations, qualitative user feedback provided by manipulation of such displays, and on-line learning of image-image distance metrics. A method for efficient, quantitative characterization and comparison of methods using simulation was also presented. The results suggest that the approach has potential for application to real-world interactive image browsing and retrieval.

Maximal-margin metric learning based on user-provided constraints performed better than ranking-SVM in this context, and in some scenarios approached the retrieval performance obtained when the user metric was known. Results suggest that such adaptation is effective even when feedback is provided through the selection of only a few images (e.g. 5) per iteration. The ability to change the query during a session can dramatically reduce the number of iterations. Interestingly, neither the dissimilarity of the initial query to the eventual retrieval result nor the mismatch of the initial system metric impacted noticeably on the number of iterations needed.

Although useful for efficient comparison of algorithms, evaluation using simulation has its limitations. One aspect that would be interesting to explore in future work would be modeling user distance metrics in ways that allow them to adapt during a browsing session. Preliminary user testing was presented here; further work will be needed to fully evaluate the proposed approach with more real users and various query scenarios.

Acknowledgements: The authors are grateful to Annette A. Ward and Ruixuan Wang for valuable discussions. This research received funding from the UK Technology Strategy Board FABRIC project, a collaboration between the University of Dundee, Liberty Art Fabrics, System Simulation, and the Victoria & Albert Museum. This research was done while J. Han was at the University of Dundee.

Reference

1. Arya, S., Mount, D.M., Netanyahu, N.S., Silverman, R., Wu, A.Y.: ‘An optimal algorithm for approximate nearest neighbor searching’, Journal of the ACM, 1998, 45, 891–923.
2. Cox, I. J., Miller, M. L., Minka, T. P., Papathomas, T. V., and Yianilos, P. N.: ‘The Bayesian image retrieval system, Pichunter: Theory, implementation, and psychophysical experiments’, IEEE Trans. on Image Processing, 2000, 9(1), 20–37.
3. Faria, F.F., Veloso, A., Almeida, H.M., Valle, E., Torres, R.d.S., Gonc¸alves, M.A., Meira, Jr., W.: Learning to rank for content-based image retrieval, Proceedings of the International Conference on Multimedia Information Retrieval, New York, USA, 2010, pp. 285–294.
4. Freund, A., Iyer, R., Schapire, R.E., Lozano-Perez, T.: ‘An efficient boosting algorithm for combining preferences’, Journal of Machine Learning Research, 2003, 4, 933–969.
5. Frome, A.,: ‘Learning Distance Functions for Examplar-based Object Recognition’, Ph.D. thesis. UC Berkeley, 2007.
6. Han, J., McKenna, S.J., Wang, R.: ‘Learning query-dependent distance metrics for interactive image retrieval’, 7th International Conference on Computer Vision Systems (ICVS), Liege, 2007.
7. He, X., Ma, W.Y., Zhang, H.J.,: ‘Learning an image manifold for retrieval’, Proceedings of the 12th Annual ACM International Conference on Multimedia, New York, 4004, pp. 17–23.
8. Hu, Y., Li, M., Yu, N.: ‘Multiple-instance ranking: Learning to rank images for image retrieval’, IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA. 2008, pp. 1–8.
9. Huang, J., Ravi Kumar, S., Mitra, M., Zhu, W., Zabih, R.: ‘Spatial color indexing and applications’, International Journal of Computer Vision, 1999, 35, 245–268.
10. Huang, W., Chan, K.L., Li, H., Lim, J.H., Liu, J., Wong, T.Y.: ‘Content-based medical image retrieval with metric learning via rank correlation,’ International Workshop on Machine Learning in Medical Imaging, 2010, pp. 18–25.
11. Joachims, T.: ‘Optimizing search engines using clickthrough data’, Proceeding of The Eighth ACM SIGKDD International Conference on Knowledge Discovery and DataMining, Alberta, Canada, 2002, pp. 133–142.
12. Lee, J.E., Jin, R., Jain, A.K.: ‘Rank-based distance metric learning: An application to image retrieval’, Proceeding of IEEE Conference on Computer Vision and Pattern Recognition, Anchorage, USA, 2008, pp.1–8.
13. Moghaddam, B., Tian, Q., Lesh, N., Shen, C., Huang, T.S.: ‘Visualization and user-modeling for browsing personal photo libraries,’ International Journal of Computer Vision, 2004, 56, 109–130.

14. Rodden, K.: ‘Evaluating Similarity-Based Visualisations as Interfaces for Image Browsing’, Ph.D. Thesis. University of Cambridge, 2001.
15. Rui, Y., Huang, T.S., Ortega, M., Mehrotra, S.: ‘Relevance feedback: A power tool in interactive content-based image retrieval’, IEEE Trans. on Circuits and Systems for Video Technology, 1998, 8, 644–655.
16. Schultz, M., Joachims, T.: ‘Learning a distance metric from relative comparisons’, Proceeding of Advances in Neural Information Processing Systems (NIPS), Berlin, Germany, 2003.
17. Smith, J.R., Chang, S.F.: ‘Automated binary texture feature sets for image retrieval’, Proceeding of IEEE International Conference on Acoustics, Speech, and Signal Processing, Atlanta, USA. 1996, pp. 2239 –2242.
18. Tong, S., Chang, E.: ‘Support vector machine active learning for image retrieval’, Proceeding of ACM Conference on Multimedia, 2001, pp. 107–118.
19. Wang, R., McKenna, S.J.: ‘Gaussian process learning from order relationships using expectation propagation’, Proceeding of International Conference on Pattern Recognition (ICPR), Istanbul, 2010, pp. 605–608.
20. Wang, R., McKenna, S.J., Han, J., Ward, A.A.: ‘Visualizing image collections using high-entropy layout distributions’, IEEE Transactions on Multimedia, 2010, 12(8), 803–813.
21. Han, J., Ngan, K., Li, M., Zhang, H.: ‘A memory learning framework for effective image retrieval’, IEEE Transactions on Image Processing, 2005, 14(4): 511-524.
22. Si, L., Jin, R., Hoi, S., Lyu, M.: ‘Collaborative image retrieval via regularized metric learning’, Multimedia System, 2006, 12(1): 34-44.
23. Wang, G., Forsyth, D., and Hoiem, D.: ‘Comparative object similarity for improved recognition with few or no examples’, Proceeding of IEEE Conference on Pattern Recogntion and Computer Vision, 2010.
24. Parikh, D., and Grauman, K.: ‘Relative Attributes’, Proceeding of International Conference on Computer Vision, 2011.